OBJECT-ORIENTED DOCUMENT ASSEMBLY SYSTEM

BACKGROUND

The following invention relates to a system for assembling documents and, in particular, to an object-oriented document assembly system for producing financial contracts.

Whenever a securities dealer executes a trade in a financial security on behalf of a client, the dealer is required by SEC regulations to send the client a financial contract confirming the trade. The document, generally called a trade confirmation, evidences all of the economic and non-economic terms of the transaction. Although trade confirmations for many types of securities are typically standardized, trade confirmations for privately negotiated securities and other swaps and derivatives must contain both specific economic terms and any special legal, credit and other non-economic terms that are applicable based upon the facts and circumstances of a particular transaction. The formats used for trade confirmations are largely based on industry standard terms and provisions, and the formats vary depending on the type of security traded. For example, with respect to swaps and derivatives, a trade association called the International Swaps and Derivatives Association, Inc. ("ISDA") publishes suggested trade confirmation formats for these types of securities.

Generally, trade confirmations are generated as follows. A trader receives a request from a client to buy, for example an option on the common shares of a public corporation ("ABC Corp."), and the trader gives the client a price for the option. If the client agrees to the price, the trader executes the trade and fills out a trade ticket that describes the economic terms of the transaction. The trade ticket is then forwarded to a back office clerk who drafts a trade confirmation appropriate for the purchase of an option on ("ABC Corp.") and that includes the economic terms of the transaction and the information regarding the client. Once the trade confirmation is drafted and reviewed, it is sent to the client for approval.

There are several drawbacks with the prior art approach to generating trade confirmations. First, because trade confirmations need to reflect the specific terms for the particular transaction, generating trade confirmations is a time-consuming and costly task, especially for high-volume dealers. Also, it is desirable for the client who wants to purchase a privately negotiated security to see the trade confirmation at the same time that the client receives a price quote for the subject security so that the client can review the economic terms before entering into the transaction. However, because of the inefficiencies of the prior art system of generating financial contracts, this is impractical. Finally, because the prior art approach requires the trader to fill out a trade ticket and a back office clerk to use the trade ticket to manually generate a trade confirmation, there is a significant risk that the resulting trade confirmation will contain errors. Accordingly, it is desirable to provide a system for generating trade confirmations that is more efficient, less costly, and less prone to errors.

Prior art systems exist for automating the assembly of documents such as invoices, receipts, timesheets, and certain correspondence. For example, U.S. Patent No. 5,893,914 to Clapp discloses an interactive computerized document assembly system using model templates. A model template is formed from a sequence of sections and has decisional options that include clause repeats, conditional clauses and questions to be answered for a particular document to be assembled. An answer index is used to store answers to the questions posed in the template. The answers corresponding to each section are merged therewith, and the merged sections are combined and displayed in proper sequence to assemble the desired document from the model template.

In U.S. Patent No. 5,729,751 to Schoolcraft, a system and a method for assembling a document and displaying information are disclosed which include a run time module coupled to document templates. The run time module processes question and manipulation codes embedded in the document templates based on answers to assembly questions, merge phase

questions, and a logic database. The question codes in the document templates prompt the system either to access associated logic records, each of which has a condition and an action, or, in the absence of an associated logic record, to ask a question. Similarly, a manipulation code triggers the system to access the associated logic record, in which case the action associated with the logic record is executed if the condition is true. As a result, the desired document is assembled from the document template based on the embedded codes and answers provided. Creation of a new document template is done by adding new and existing codes to a master document and adding any new codes to the corresponding databases.

In summary, the prior art document assembly systems use templates to generate documents of a particular document type. Each template includes a variety of clauses that are used to construct a document of the corresponding type. Associated with each template is a plurality of questions pertinent to the construction of the desired document. The answers to the questions are used to determine what clauses should be eliminated, added, or repeated in the final document. Thus, a single template is used to assemble documents which differ in content (as determined by the particular clauses selected) but are comparable in presentation (as determined by the particular template).

The prior art template-based document assembly systems have several disadvantages. First, in the prior art systems, a unique template must be created for each document type to be assembled. Furthermore, any significant changes to a document type may require the creation of a new template incorporating those changes. Because the skill level and difficulty associated with creating a new template varies from system to system, the process of adding new templates in the prior art systems is often time-consuming and error-prone.

Also, the prior art template-based systems are difficult to maintain. If it becomes necessary to alter any information common to multiple templates, then the changes must be made to each template independently. For instance, if a new law requires a change to a

provision that is common to many different financial contract templates, then all templates that contain provisions affected by the new law must be separately updated to reflect the changes. Having to manage which templates to update as field information changes makes the prior art systems cumbersome to maintain.

Finally, because the prior art systems use a separate template for each document type to be assembled, storage requirements for those systems increase in proportion to the number of document types desired.

A prior art template-based document assembly system for generating trade confirmations, called DocSolution for Swaps and Derivatives, is provided by Documentum (www.documentum.com). The DocSolution system focuses on the assembly of trade confirmations for trades involving swaps and derivatives. However, because the DocSolution system is template-based, it suffers from the same deficiencies as template-based systems generally.

Accordingly, it is desirable to provide a document assembly system in which new document types are easily added, inefficiency is reduced, updates to the system are efficiently managed, and storage requirements are reduced.

SUMMARY OF THE INVENTION

The present invention is directed to a system and a method of object-oriented document assembly that overcome the deficiencies of the prior art. According to the present invention, an object-oriented system for assembling a document is provided that includes a plurality of terms, a plurality of objects, and a plurality of grammar lines. Each of the plurality of objects includes an object tag. At least one of the plurality of objects includes at least one of the plurality of terms. Each of the plurality of grammar lines includes a condition and an instruction. At least one of the conditions includes at least one of the plurality of terms. When the condition of one of the plurality of grammar lines is true, then the

instruction associated with that condition is executed, thereby assembling at least a portion of the document.

The plurality of terms is the data underlying the document to be assembled. In the case of a financial contract, the terms would encompass the economic and non-economic terms of the transaction. Each term belongs to a specific category of information, and each category of information is represented by a trade term variable. The plurality of objects may contain information in the form of fixed text, variable text, or visual images. Objects of a trade confirmation generation system may contain, for example, the name and graphical logo of the company issuing the confirmation, as fixed text and a visual image respectively, with the date of the trade as variable text. Variable text incorporates trade term variables that assume values derived from the plurality of terms. Inclusion of a term within an object is achieved through use of the trade term variable corresponding to the term. At least one of the plurality of objects incorporates a trade term variable having a value determined by one of the plurality of terms, such as the name of the transactional counterparty in a trade confirmation. Similarly, at least one of the plurality of objects contains information for inclusion in the document, such as a standard legal disclaimer. Through combining one or more of the plurality of objects, a sequence of information is assembled that, when complete, is the assembled document.

The plurality of grammar lines guides the assembly of the document. Each of the plurality of grammar lines has a condition and an instruction. The condition states the circumstances under which the associated grammar line will be executed. The instruction directs the system as to how to proceed in assembling the document, either by directing the system to move to another grammar line, or by instructing the system to append an object to the sequence of information. Moving between grammar lines is accomplished by the use of grammar tags to identify the grammar lines in conjunction with the use of instructions

containing the grammar tags to which the system is directed to move. Appending an object causes information, one or more terms (via the use of trade term variables), graphics, or a combination thereof to be included in the document. To determine whether to execute the instruction associated with a grammar line, the associated condition is tested. When the condition of a grammar line is true, each instruction associated with that grammar line is executed, thereby advancing the assembly of the document. Executing an instruction may include processing another grammar line and/or selecting an object which contains information to be inserted in the document. After all of the required instructions have been executed, the document is fully assembled, containing the sequence of desired information based on the plurality of terms associated with a particular transaction. It is envisioned that the system will comprise a computer system that will cause the document to be assembled and be readable/modifiable using a word processor application.

The object-oriented architecture of the document assembly system of the present invention is designed to be an open, flexible system able to produce documents of any variety, ranging from the highly-standardized to highly-specialized. For each financial transaction, a document is independently assembled from a library of objects based on the assembly rules expressed in the grammar lines and in view of the terms of the transaction to be documented. Consequently, the system of the present invention does not require the use of fixed templates for each document type, as in the prior art systems. Furthermore, because an object need only be changed once for the change to be available to all documents generated by the system, the system of the present invention is simpler to maintain than the prior art document assembly systems.

The invention accordingly comprises the features of construction, combination of elements and arrangement of parts that will be exemplified in the following detailed disclosure, and the scope of the invention will be indicated in the claims. Other features and

advantages of the invention will be apparent from the description, the drawings and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

For a fuller understanding of the invention, reference is made to the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a block diagram of the object-oriented document assembly system of the present invention;

FIG. 2 is a table view of a portion of the transaction file of FIG. 1 according to an exemplary embodiment;

FIGS. 3A-3C are objects contained in the object library of FIG. 1 according to an exemplary embodiment;

FIG. 4 is a table view of a portion of the grammar of FIG. 1 according to an exemplary embodiment;

FIG. 5 illustrates a portion of terms of a transaction from the transaction file of FIG. 1 according to an exemplary embodiment;

FIG. 6 is a table view of a portion of the grammar of FIG. 1 according to an exemplary embodiment;

FIG. 7 is a table view of a portion of the grammar of FIG. 1 according to an exemplary embodiment;

FIG. 8 is a table view of a portion of the grammar of FIG. 1 according to an exemplary embodiment;

FIG. 9 is an object contained in the object library of FIG. 1 according to an exemplary embodiment;

FIG. 10 is a document assembled by the object-oriented document assembly system of FIG. 1 according to an exemplary embodiment; and

FIG. 11 is a table view of a portion of the grammar of FIG. 1 illustrating a default tag according to an exemplary embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to FIG. 1, there is shown a document assembly system 100 of the present invention. System 100 includes a transaction file 101 that stores terms that describe each of a plurality financial transactions 101(l)-101(n) performed by a trading system 1001. System 100 also includes an object library 106 that includes a plurality of objects 106(l)-106(n). Each of objects 106(l)-106(n) have an object tag 107 and an object body 108. Also included in system 100 is a grammar 113. Grammar 113 includes a plurality of grammar lines 113(l)-113(n) with each of grammar lines 113(l)-113(n) having a grammar tag 128, an instruction 116 and a condition 115. Applying grammar 113 in a manner described below will generate a document 154 that reflects the terms and conditions associated with a particular financial transaction performed by trading system 1001.

Referring now to FIG. 2, there is shown a table view of a portion of transaction file 101 according to an exemplary embodiment. Each financial transaction 101(1)-101(25) listed in transaction file 101 is described by a plurality of trade terms 104. For example, term 104(1,5) is the date financial transaction 101(1) took place, term 104(1,12) is the name of the instrument that was the subject of transaction 101(1) and term 104(1,6) is the customer account number on whose behalf transaction 101(1) was entered. Transactions file 101 of FIG. 2 contains trade terms 104 for OTC options transactions so transaction file 101 includes option specific trade terms such as the strike price (term 104(1,13)), the option expiration date (term 104(1,14)) and the option style (term 104(1,10)). Transaction file 101, however, is not limited to options transactions but can include terms relating to a transaction in any type of financial instrument (e.g., swaps) or any type of financial contract or master agreement (e.g., ISDA Master Agreements).

Referring now to FIGS. 3A-3C, there are shown object types contained in object library 106 according to an exemplary embodiment. For example, in FIG. 3A object 106(1) includes object tag 107(1) containing the label [strike_default]. Object body 108(1) of object 106(1) includes a fixed text portion 3108 as well as a variable portion 3109. Variable portion 3109 includes a trade term variable 105(13), that represents the strike price of the option and a trade term variable 105(40), that represents the currency of the strike price. When object 106(1) is applied by system 100 during the trade confirmation generation process, system 100 will insert fixed text portion 3108 into the document. System 100 will also replace trade term variables 105(13) and 105(40) with the corresponding terms associated with the particular transaction. Thus, for transaction 101(1) involving an option having a strike price of 40 (term 104(1,13)) and a strike price currency of US$, the following line is inserted in the trade confirmation document:

Strike Price: US$40

Object 106(2) shown in FIG. 3B includes object tag 107(2) containing the label [disclaim_ftse]. Object body 108(2) of object 106(2) contains only fixed text that, in this particular example, is a disclaimer clause. If object 106(2) is applied by system 100 based on particular transaction terms, the disclaimer clause contained in object body 108(2) is placed in the resulting document.

Object 106(3) shown in FIG. 3C includes object tag 107(3) containing the label [sign_msil]. Object body 108(3) contains the fixed text portion of a signature block to be placed at the bottom of the resulting document as well as trade term variables 105(17)-105(20) that will be replaced with the corresponding terms 104 associated with a particular financial transaction 101(x) that provide the detail to be included in the signature block.

In addition to the embodiments described above, object body 108 may contain any other text, graphic, variable or other information desirable for generating a confirmation for a particular financial transaction or any other desired document. For example, object body 108 may include a company logo, a letter head with date and address and a letter body describing the transaction including the number, price, style and price of the subject instrument. It will be obvious to one of ordinary skill that object body 108 can include any information type that would be desirable to include in the document to be assembled by system 100.

Referring now to FIG. 4, there is shown a table view of a portion of grammar 113 according to an exemplary embodiment of the present invention. Each grammar line 113(x) includes grammar tag 128(x) which is used to identify the particular grammar line 113(x). In particular, each of grammar lines 113(1)-113(10) have a <start> label as grammar tags 128(1)-128(10), respectively, which indicates that grammar lines 113(1)-113(10) are applied first by system 100.

Condition 115(x) associated with grammar line 113(x) is generally an expression that includes at least one trade term variable 105 and, based on terms104(x,y) associated with transaction 101(x), evaluates to either a true or false statement. For example, condition 115(2) of grammar line 113(2) tests trade term variable 105(21) ([CONFIRMSTYLE]) to determine if it contains the trade term "CELTS" (which would indicate that a particular trade confirmation style is desired) and trade term variable 105(6) ([CPACCT]) to determine if it contains the trade term "62B0686" (a particular account number). If both [CONFIRMSTYLE] contained "CELTS" and [CPACCT] contained "62B0686", then condition 115(2) would be a true expression and instruction 116(2) would be executed. It will be obvious to one of ordinary skill in the art that conditions 115(1)-115(n) may include any number and combination of trade term variables 105 combined with any suitable logic or mathematical operators to form any desired expression. In certain cases, some of conditions

115(1)-115(n) may not include trade term variable 105 and will therefore always evaluate as a true statement.

As is the case for grammar lines 113(1)-113(10) in the embodiment shown in FIG. 4, a portion of grammar lines 113(x) can have identical grammar tags 128. If that occurs, in all but one circumstance (which will be described below), conditions 115 associated with those of grammar lines 113(x) having identical grammar tags 128 are mutually exclusive. So, for example, conditions 115(1)-115(10) are all mutually exclusive – i.e., only one of conditions 115(1)-115(10) can be true for a given set of terms 104. In this situation, system 100 will determine which of conditions 115(1)-115(10) is true and execute instruction 116 associated therewith.

Instruction 116(x) associated with grammar line 113(x) contain either one or more of grammar tags 128(x), one or more of object tags 107(x), or a combination of both. For example, instruction 116(3) contain four grammar tags (<header>, <msaddress>, <split_corpcov> and <sign>) and one object tag ([fax_default]). When instruction 116(3) is executed by system 100 (in the event condition 115(3) is true), system 100 will determine which of the conditions associated with grammar lines having a <header> grammar tag is true and execute the associated instruction. System 100 will then test the conditions associated with grammar lines having a <msaddress>, <split_corpcov>, respectively, and execute the associated instruction for the grammar lines having true a condition. System 100 will then insert the object body of the [fax_default] object into document 154. Thereafter, system 100 will test the conditions associated with grammar lines having the grammar tag <sign> and execute the associated instruction for the <sign> grammar line having a true condition.

Referring now to FIGS 5-10, there is shown a sequence of screen shots that demonstrate the operation of system 100 according to an exemplary embodiment. FIG. 5 is a screen shot of system 100 showing terms 104(26,x) associated with financial transaction

101(26) that is an Out of the Money Expiration of an OTC Equity Option. The first step of the trade confirmation generation process performed by system 100 is to test terms 104(26,x) of transaction 101(26) against the conditions of all grammar lines 113(x) having a <start> grammar tag, a portion of which are shown in FIG. 6. In this case, condition 115(11) of grammar line 113(11) is "true" based on terms 104(26,x) of transaction 101(26) and, as a result, instruction 116(11) is executed.

Instruction 116(11) of grammar line 113(11) contains three elements -- <header>, <msaddress> and <split> -- each of which is a grammar tag associated with other grammar lines in grammar 113. It is through the application of these grammar tags - and the execution of any subsequent instructions associated with these "nested" grammar tags - that system 100 assembles a document type incorporating terms 104(26,x) relevant to transaction 101(26).

With respect to executing instruction 116(11), system 100 will first process the grammar lines having a <header> grammar tag. Referring now to FIG. 7, there are shown grammar lines 113(12)-113(27) that each have a <header> grammar tag. System 100 will then determine which of conditions 115(12)-115(27) is true. Because it is required that (with one exception to be discussed below) each condition 115(x) associated with a grammar tag that is not unique be mutually exclusive of the other such conditions having the same grammar tag, only one of conditions 115(12)-115(27) will be true based on terms 104(26,x) associated with transaction 101(26). In this particular example, condition 115(16) is true and, as a result, instruction 116(16) will be executed by system 100.

Instruction 116(16) includes [header_msil] (object tag 107(4)) which has associated therewith object body 108(4). System 100 executes instruction 116(16) by inserting object body 108(4) into the trade confirmation document to being generated. Object body 108(4) includes trade term variables 105(3) and 105(6) which system 100 will replace with the corresponding trade terms included in terms 104(26,3) and 104(26,6).

After finding the one of grammar lines 113(12)-113(27) having a <header> grammar tag for which the corresponding condition is true and executing the associated instruction, system 100 then returns to grammar line 113(11) to execute the next element of instruction 116(11) – the <msaddress> grammar tag.

Referring now to FIG. 8, there are shown grammar lines 113(28)-113(36) that each have <msaddress> as a grammar tag. As with grammar lines 113(12)-113(27) having <header> as a grammar tag, system 100 will determine which of conditions 115(28)-115(36) is true and execute the associated instruction. In this embodiment, condition 115(34) is true and instruction 116(34) is executed. Execution of instruction 116(34), which includes object tag 107(5), results in the insertion of object body 108(5), shown in FIG. 9, into the trade confirmation document being assembled.

After executing instruction 116(34) associated with grammar line 113(34) having an <msaddress> grammar tag, in a similar manner system 100 will then process the <split> grammar tag included in instruction 116(11) and execute in sequence any instructions that arise therefrom. After system 100 completes the execution of instruction 116(11) using the grammar of this embodiment, document 154, shown in FIG. 10, is generated. Note that in the process of compiling document 154 based on grammar 113 and terms 104(26,x) of transaction 101(26), system 100 replaces trade term variables 105(3) and 105(6) with the corresponding trade terms 104(26,3) and 104(26,6), repectively. Likewise, system 100 will replace other trade term variables with the corresponding terms associated with the particular transaction.

Referring now to FIG. 11, there is shown grammar line 113(37) in which a default tag box 127(37) is checked and condition 115(37) is a null condition (always true). Generally, all of grammar lines 113(x) having identical grammar tags will have mutually exclusive conditions so that only one of the condition is true for a given set of terms. However there

are situations where it is desirable to have a default grammar line that system 100 will execute if none of the other grammar lines having the same grammar tag have a condition that is true. For example, grammar lines 113(37) and 113(38) each have identical grammar tags 128(37) and 128(38), respectively. In operation, system 100 first tests condition 115(38) of grammar line 113(38) to determine if it is true – in this case, whether the particular financial for which a confirmation is being generated is for counterparty account 62B0094. If it is, then system 100 will execute instruction 116(38) that includes an object that accounts for special addressing requirements for the specific client. If, on the other hand, condition 115(38) is false, which indicates that no special addressing requirements are necessary, then system 100 will execute instruction 116(37) associated with grammar line 113(37) that has default tag 127(37) checked. In this case, instruction 116(37) includes an object that provides a generic addressing format. Thus, default tag 127(37) eliminates the need to make condition 115(37) of grammar line 113(37) both true and mutually exclusive of condition 115(38) of grammar line 113(38). In this way, the use of default tag 127 enables system 100 to accommodate the specific requirements of particular clients and situations while at the same time having a default mechanism to serve the general client population.

Accordingly, by using system 100 of the present invention, a document assembly system is provided that overcomes the deficiencies of the prior art systems. Unlike the prior art systems that are template-based and that require a unique template for each document type to be assembled, system 100 uses grammar 113 consisting of grammar lines 113(x) to construct all desirable document types. When a new document type is desired, a new <start> grammar line is inserted into grammar 113 that causes a sequence of instructions contained in grammar 113 to be executed that results in the document being generated. Thus, the prior art drawback of creating an entirely new document template for each new document type is overcome.

In addition, because, in the present invention, all the instructions used to create the various documents types are contained in grammar 113, the need to have separate templates for each document type containing the required fields, as is the case in the prior art template-based systems, is eliminated. Many different document types may utilize several common objects and/or grammar lines, but each grammar line needs to appear only once. Thus, by eliminating the redundancy inherent in the template-based approach, storage requirements for system 100 are reduced.

System 100 also greatly simplifies the updating of information to be included in documents to be generated. While the prior art template-based systems generally require changes to each template that use a particular field that is to be updated, in system 100 of the present invention only the particular object that contains the information to be changed needs to modified. Because all of objects 106(x) contained in object library 106 are unique, any change to a piece of information controlled by a particular object need only be made once. Also, because system 100 reuses objects 106(x), as necessary, in the process of assembling different document types, the changes made to a particular object affect any grammar line sequence that uses the modified object to generate a particular document. Accordingly, document assembly system 100 of the present invention is easily maintained.

Although system 100 was described above in relation to the assembly of trade confirmation documents and financial contracts, it will be obvious to one of ordinary skill that system 100 can be applied to generate other types of documents as well including, but not limited to, invoices, correspondence and memorandum.

A number of embodiments of the present invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims. Because certain changes may be made in the construction set

forth above without departing from the spirit and scope of the invention, it is intended that all matter contained in the above description or shown in the accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

It is also to be understood that the following claims are intended to cover all of the generic and specific features of the invention herein described, and all statements of the scope of the invention, which, as a matter of language, might be said to fall therebetween.